

Carrera de Data Scientist

Juan Morales Volosín

Ha realizado y completado con éxito su carrera en Coderhouse.
La duración fue de 40 semanas, cumpliendo todos los
requisitos académicos exigidos.

24 de octubre de 2024



Alejandra Vatrano

Directora académica



Christian Patiño

CEO y Co-founder en Coderhouse

Contenidos

En las siguientes diapositivas, y separados por curso, presento un pantallazo de los trabajos finales correspondientes. Para estos utilicé Power BI, Python (con 'pandas') y Google Colaboratory.

En el repositorio, adicionalmente a estos trabajos se encuentran otros realizados durante la cursada.

Para acceder a todos los archivos, hacer clic en este [enlace al repositorio](#).

Data Analytics

Juan Morales Volosín

Ha realizado y completado con éxito su curso en Coderhouse.
La duración fue de 46 horas dictadas a lo largo de 12 semanas,
cumpliendo todos los requisitos académicos exigidos.

12 de marzo de 2024



Pablo Guzzi

Director de la Carrera de Data
Chief Data & Analytics Officer en Ualá

Certificado por



Christian Patiño

CEO y Co-founder en Coderhouse

Estudio Sociodemográfico de la Ciudad Autónoma de Buenos Aires 2019

Por
Juan Ignacio Morales Volosín

INICIO

LABORAL

EDUCACIÓN

SALUD

GLOSARIO

Botones
Interactivos





Estudio Sociodemográfico de la Ciudad Autónoma de Buenos Aires 2019

Inicio

La Ciudad

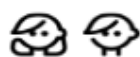
Población Total Superficie (km²)

239712

21.7

Información general sobre la Ciudad de Buenos Aires, y sobre la muestra estudiada: la Encuesta Anual de Hogares de 2019. En las próximas hojas se analiza únicamente la muestra.

Por Sexo



Borrar

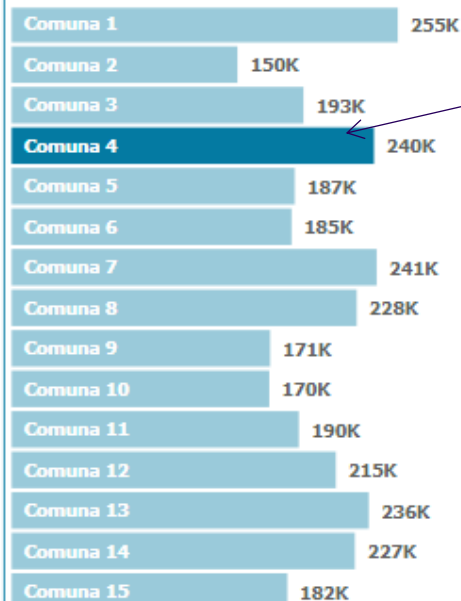


Por Hogar

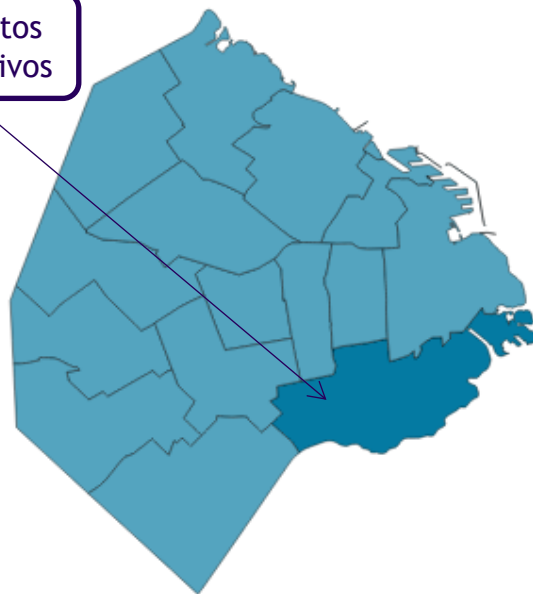


Filtros

Distribución de Población



Elementos interactivos



Muestra Encuestada

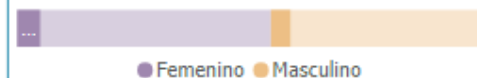
Personas

1.3K

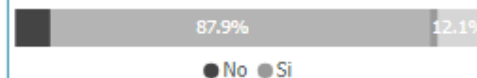
Hogares

494

Personas - Muestreo por Sexo



Hogares - Muestreo en Villas de Emergencia



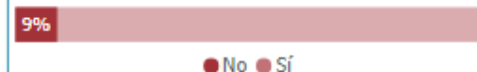
Hogares Laboralmente Activos

Al menos un miembro trabaja



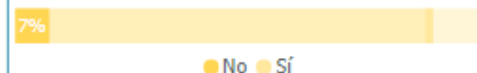
Hogares con Analfabetismo

Al menos un miembro adulto jamás estudió



Hogares sólo con Salud Pública

Ningún miembro tiene acceso a otro sistema



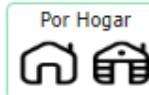
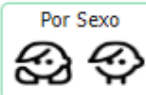


Estudio Sociodemográfico de la Ciudad Autónoma de Buenos Aires 2019

Trabajo

Tasa de
Desempleo
5.93%

Tasa de
Participación
81.54%



Información del nivel de ocupación en la Ciudad de Buenos Aires, indicando promedios de ingresos por persona y por hogar. También se detalla cuánta gente se encuentra en actividad, o qué clase de ocupación desarrollan.

Personas

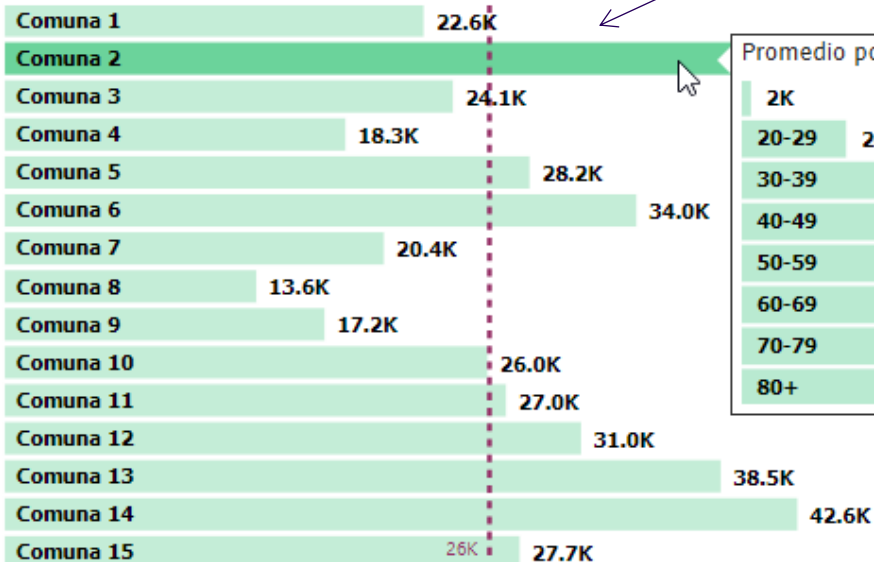
Hogares

Filtros

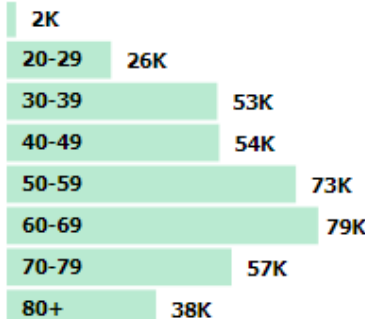
Elementos
interactivos

Ingreso Promedio por Persona

Y Línea de Promedio General

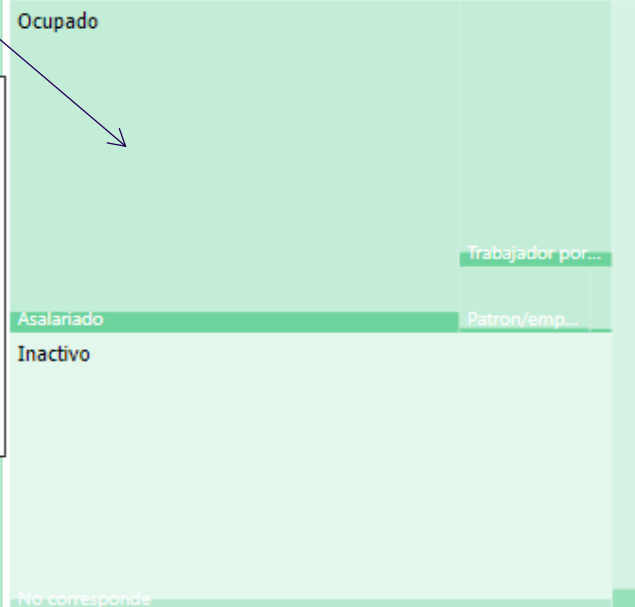


Promedio por Rango Etario



Tooltip

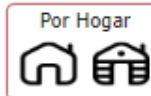
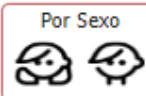
Distribución por Estado y Categoría Laborales





Estudio Sociodemográfico de la Ciudad Autónoma de Buenos Aires 2019

Educación



Se presenta información sobre el nivel educativo en la Ciudad, mostrando cuántos años de escolaridad cursaron sus ciudadanos. Asimismo, se analiza a fondo qué tipos de estudios cursaron, cursan, y en qué tipos de instituciones.

Personas
Estudiando
30.84%

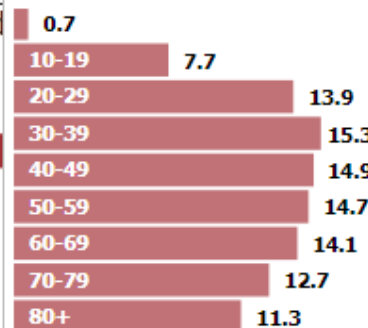
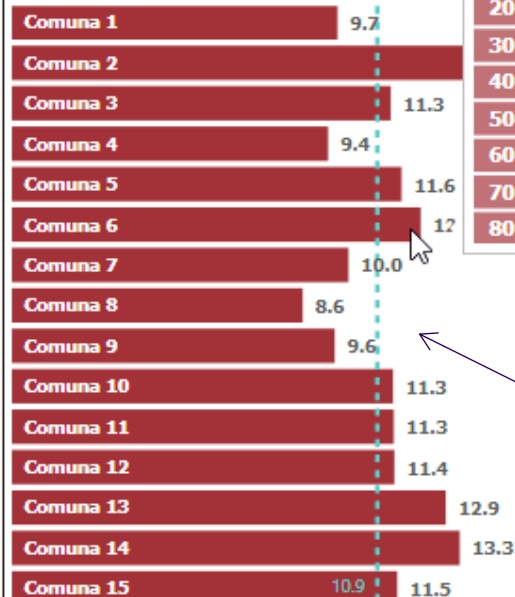
Personas
Analfabetas
0.30%



Promedio por Rango Etario

Promedio de Años de Escolaridad

Por Comuna y Total de la Muestra



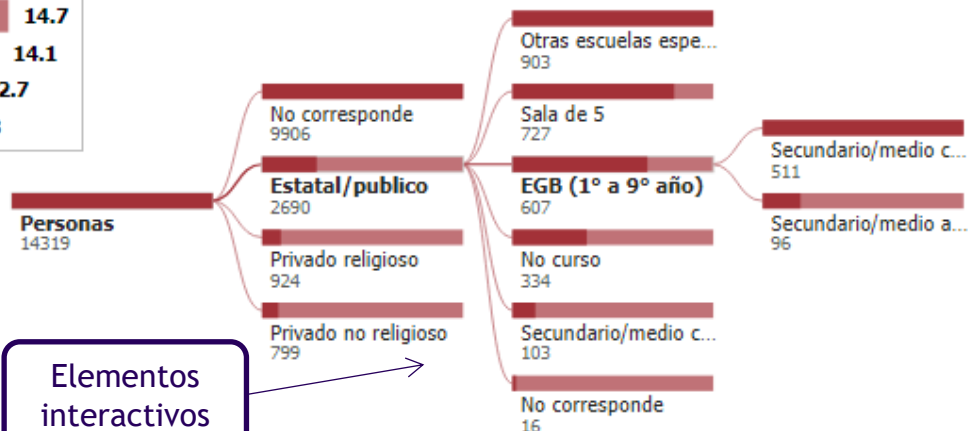
Composición del Nivel Educativo por Persona

Desglose por Sector y Niveles Alcanzado y en Curso

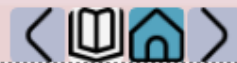
Sector Educativo

Nivel ya Alcanzado

Nivel en Curso



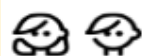
Elementos
interactivos



Estudio Sociodemográfico de la Ciudad Autónoma de Buenos Aires 2019

Salud

Por Sexo



Borrar



Por Hogar



Presentación de la información sobre el acceso a salud por parte de los habitantes de la Ciudad de Buenos Aires. Se detalla la naturaleza del mismo por hogares, más profundamente por persona, y se analiza pertenencia a cada sistema.

Hogares y su Acceso a Salud:

Público	Pago	Mixto
632	4472	744

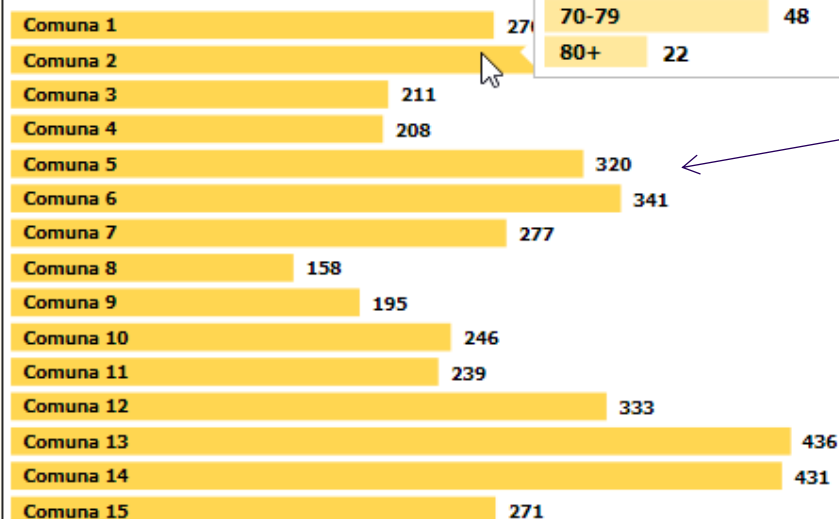
Público

O. Sociales

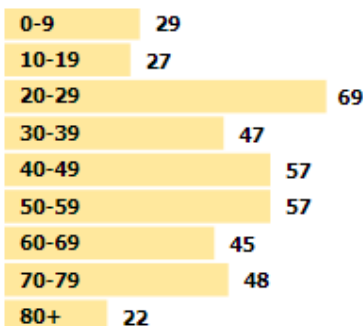
Ot

Personas y su Acceso a Salud

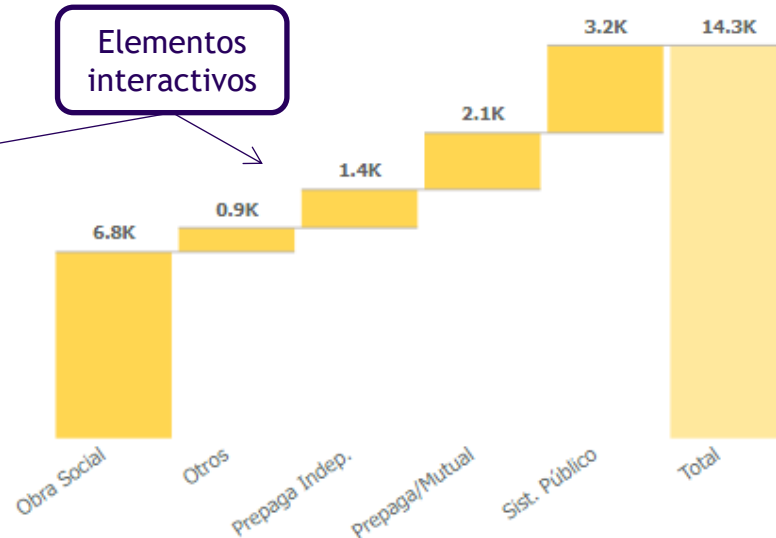
Vía Prepagas, Sistemas Mixtos u Otros



Personas por Rango Etario



Personas Utilizando Cada Sistema



Elementos interactivos



Data Science I: Fundamentos para la Ciencia de Datos

Juan Morales Volosín

Ha realizado y completado con éxito su curso en Coderhouse.
La duración fue de 38 horas dictadas a lo largo de 10 semanas,
cumpliendo todos los requisitos académicos exigidos.

21 de mayo de 2024



Tamara Drajer

Coderhouse
Head de operaciones académicas

Certificado por



Christian Patiño

CEO y Co-founder en Coderhouse



Carga de Datos

Exploración General

Exploración Variable a Variable

Índice, o 'id'

Variable 'name'

Variable 'host_id'

Variable 'host_name'

Variables
'neighbourhood_group',
'neighbourhood'

Variables 'latitude' y
'longitude'

Variable 'room_type'

Variable 'price'

Distribución de Precio
por Distrito

Relación de precios y
cantidad de noches, y
precios y
disponibilidad anual

✓ Precios de Airbnb en Nueva York 2019

✓ Objetivo

Se busca comprender los datos mencionados mediante el uso de bibliotecas de Python, como Pandas, Matplotlib y Seaborn, entre otras.

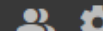
Una vez logrado esto, se apunta a entrenar un modelo de aprendizaje automático que, en última instancia, sea capaz de predecir el precio de un Airbnb basándose en otros campos del dataset.

✓ Introducción

✓ Contexto Empresarial

En su búsqueda de constante mejora, los ejecutivos de Airbnb han decidido buscar una forma de asistir a los nuevos anfitriones recientemente registrados en la plataforma. De esta forma, esperan facilitarles el camino haciendo más conveniente el alquiler de una de sus propiedades, o parte de ella, como una habitación. Para simplificar, se lo mencionará como "un Airbnb".

Teniendo esto en cuenta, se han acercado a su equipo de científicos de datos a fin de encargarles diferentes tareas.



Carga de Datos

Exploración General

Exploración Variable a Variable

Índice, o 'id'

Variable 'name'

Variable 'host_id'

Variable 'host_name'

Variables'nghbhood_group',
'nghbhood'Variables 'latitude' y
'longitude'

Variable 'room_type'

Variable 'price'

En Python 'wordcloud',
separando pertenencia a
cada distrito mediante color.

Relación de precios y
cantidad de noches, y
precios y
disponibilidad anual

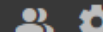


Distritos



Barrios





Carga de Datos

Exploración General

Exploración Variable a Variable

Índice, o 'id'

Variable 'name'

Variable 'host_id'

Variable 'host_name'

Variables
'neighbourhood_group',
'neighbourhood'**Variables 'latitude' y
'longitude'**

Variable 'room_type'

Variable 'price'

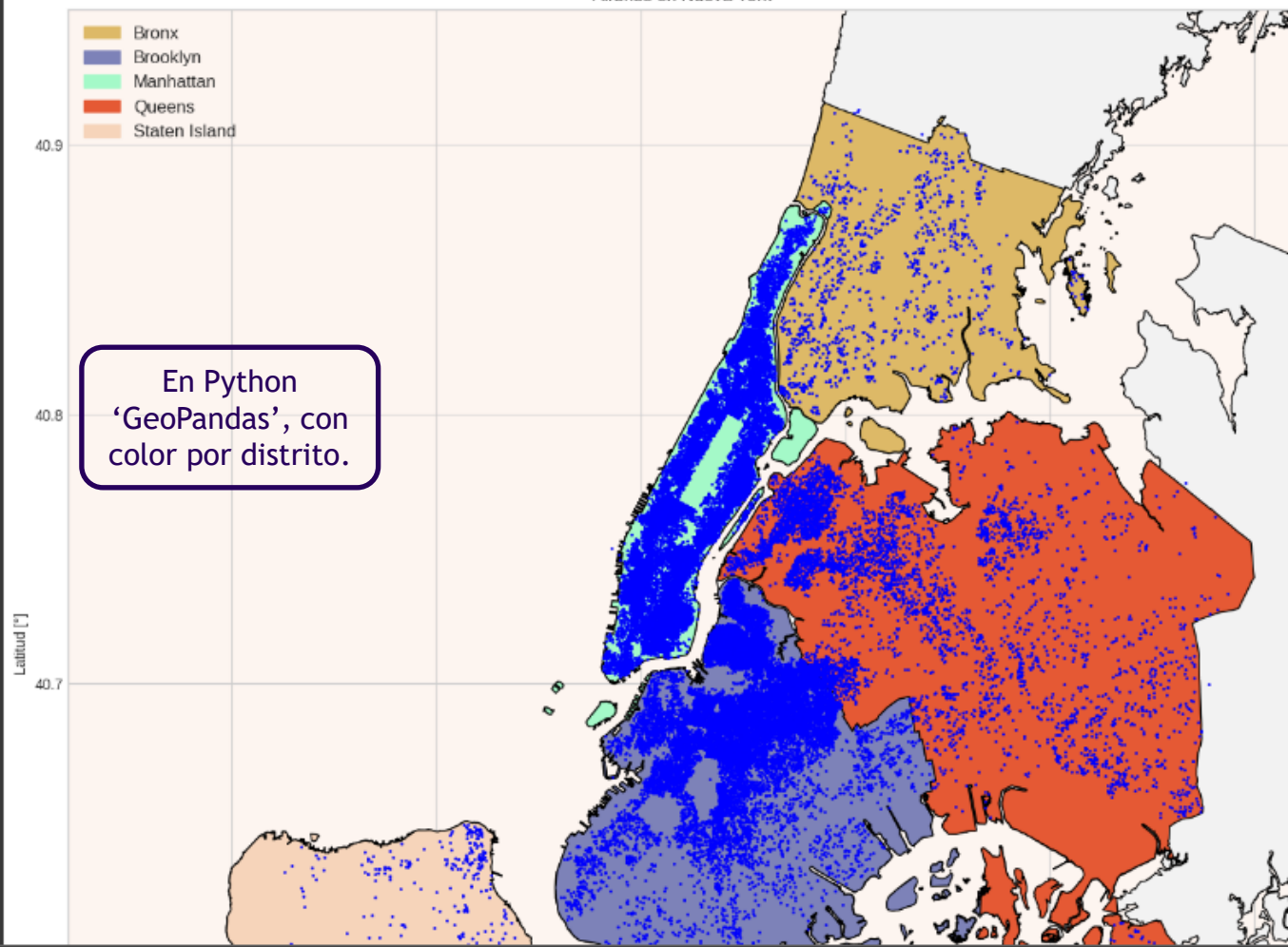
Distribución de Precio
por DistritoRelación de precios y
cantidad de noches, y
precios y
disponibilidad anualUbicación Geográfica de los
Airbnbs en Nueva York

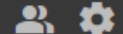
Table of contents



+ Code + Text

Connect

Gemini



- Variable 'minimum_nights'
- Variable 'last_review'
- Variable 'no_of_reviews'
- Variable 'reviews_month'
- Variable 'calc_hlstngs_count'
- Variable 'availability_365'

Primeros Comentarios

Relaciones Entre Variables

Implementación de Modelos

Bibliotecas

Regresión Lineal

Características Polinomiales

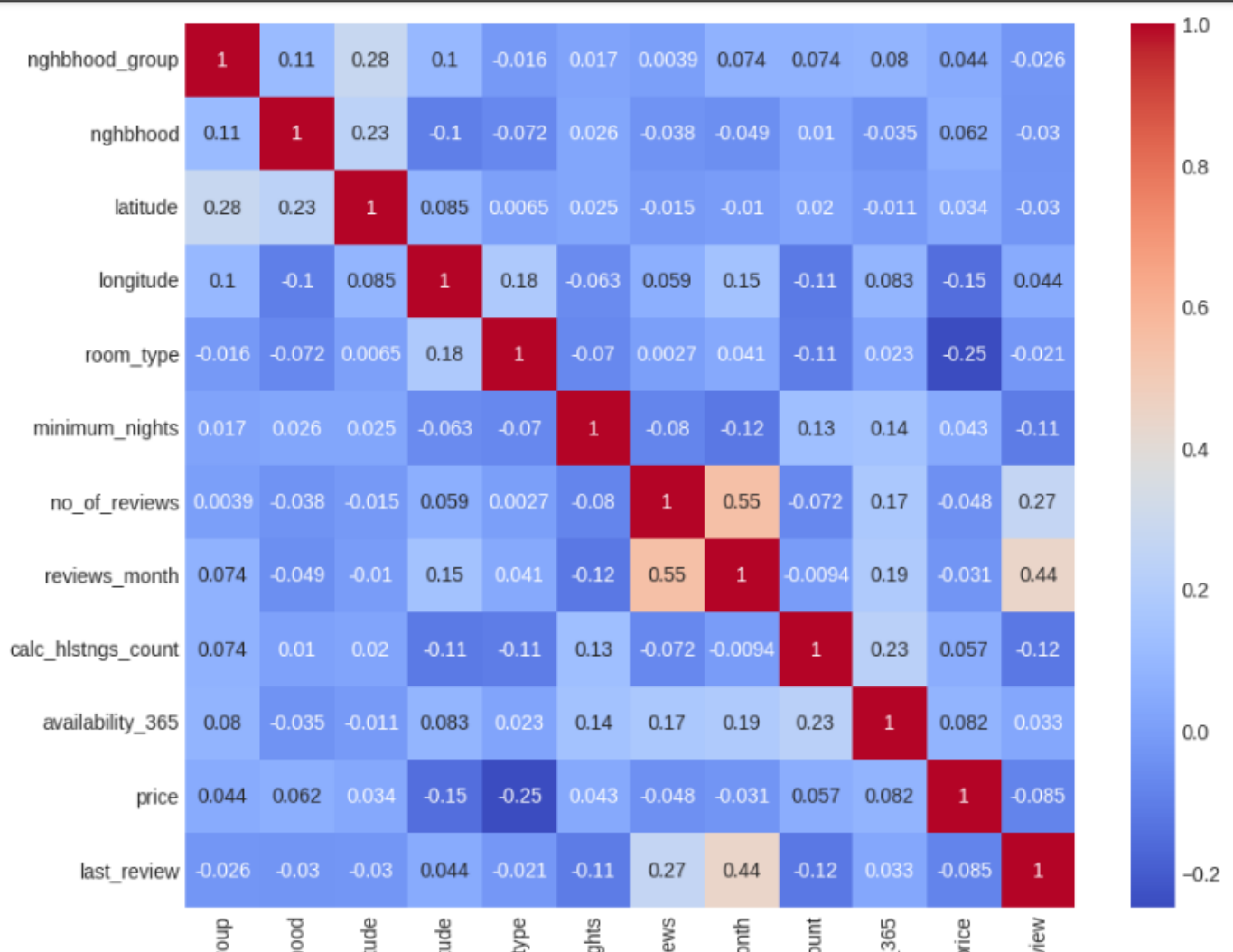
Cuadrático

Cúbico

k Nearest Neighbors

Comparación de Métricas

En Python 'seaborn',
con escala bicromática
según signo.





- kNN SA
- Comparación de Métricas SA
- Selección de Características
- Regresión Lineal SC
- Características Polinomiales SC
- Cuadrático
- Cúbico
- kNN SC
- Comparación de Métricas SC**
- Conclusiones
- Predicción de la Variable 'price'
- Insuficiencia del Conjunto de Datos

```
print('Estas son las metricas obtenidas tras aplicar seleccion de caracteristicas:')  
print(df_sc.round(4).to_string(max_colwidth=20))
```

Estas son las métricas que se obtuvieron originalmente:

	lr	pf2	pf3	knn
Train MSE	57396.0756	55788.6558	55788.6558	42673.3777
Test MSE	32889.1487	31977.6365	31977.6365	35656.5682
Train MAE	76.1526	72.3069	72.3069	61.5674
Test MAE	72.6866	68.6963	68.6963	68.0970
Train RMSE	239.5748	236.1962	236.1962	206.5754
Test RMSE	181.3537	178.8229	178.8229	188.8295
Train R2	0.0845	0.1102	0.1102	0.3194
Test R2	0.1245	0.1488	0.1488	0.0508

Estas son las métricas obtenidas tras eliminar anómalos de 'price':

	lr	pf2	pf3	knn
Train MSE	2583.9705	2223.9443	2223.9443	1956.2007
Test MSE	2485.0768	2108.6147	2108.6147	2013.6765
Train MAE	38.1308	34.6078	34.6078	32.0559
Test MAE	37.2018	33.7712	33.7712	32.8213
Train RMSE	50.8328	47.1587	47.1587	44.2290
Test RMSE	49.8505	45.9197	45.9197	44.8740
Train R2	0.4472	0.5242	0.5242	0.5815
Test R2	0.4506	0.5338	0.5338	0.5548

Estas son las métricas obtenidas tras aplicar selección de características:

	lr	pf2	pf3	knn
Train MSE	2590.1583	2339.2074	0.0001	2082.1507
Test MSE	2491.8563	2231.4617	0.0001	2077.8014
Train MAE	38.1465	35.7144	0.0001	32.9605
Test MAE	37.2042	34.8276	0.0001	33.0250
Train RMSE	50.8936	48.3654	0.0001	45.6306
Test RMSE	49.9185	47.2383	0.0001	45.5829
Train R2	0.4459	0.4996	0.0001	0.5546
Test R2	0.4491	0.5067	0.0001	0.5406

Data Science II: Machine Learning para la Ciencia de Datos

Juan Morales Volosín

Ha realizado y completado con éxito su curso en Coderhouse.
La duración fue de 62 horas dictadas a lo largo de 16 semanas,
cumpliendo todos los requisitos académicos exigidos.

24 de septiembre de 2024



Tamara Drajner

Coderhouse
Head de operaciones académicas




Christian Patiño

CEO y Co-founder en Coderhouse



A Study of Objects in Earth Orbit for a Space Debris Removal Company




 Objective

 Introduction

 Business Context

 Analytical Context


 Hypothesis

 Data Description


 Source

 Variables


 Libraries

 Custom Functions

 Configuration

 Reading the Database

 Worlds in the Solar System

 Object Catalogs

 Main Catalogs

 Dataset Consolidation



A Study of Objects in Earth Orbit for a Space Debris Removal Company



Course:

Data Science II: Machine Learning para la Ciencia de Datos

Commission #60955



Student:

Morales Volosín, Juan Ignacio



Professor:

Paredes, Julio



Tutor:

Reale, Victor Adrián



Objective

This work aims to analyze the data presented in Jonathan McDowell's General Catalog of Artificial Space Objects (GCAT) using the Python programming language and libraries such as NumPy, Pandas, and Bokeh, among others. The analysis will particularly place the

Table of contents



+ Code + Text

Connect

Gemini



Selector de variable y tipo de gráfico en Python 'ipywidgets', gráficos en 'wordcloud' y 'matplotlib/seaborn'.

```
[ ] # Display the widgets
display(wdg_sel_row)
display(wdg_count_display)
display(output)
```

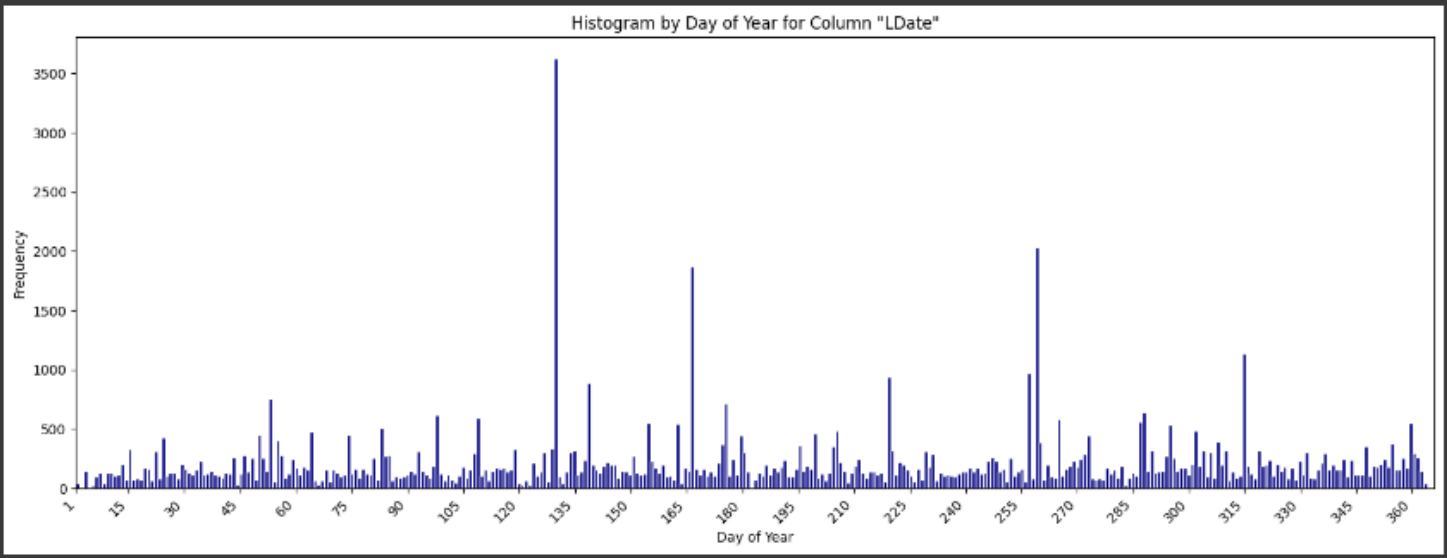


Column: LDate

Graphic: YearDayplot

Plot

Count: 1970-	Unique:	Top:	Freq:	Mean: 1995-	Std:	Min: 1957-10-	25%: 1979-	50%: 1993-	75%: 2015-	Max: 2024-
01-01 00:00:00	NaT	NaT	NaT	09-29	NaT	04 00:00:00	10-31	12-18	07-24	12-18
				08:46:44			00:00:00	00:00:00	00:00:00	00:00:00



Select the column, then the graphic (if needed), and hit 'Plot'.

Table of contents

Span

📁 Saving the Consolidated Dataset

🔍 Exploratory Data Analysis

📊 General Analysis of the DataFrame

🇺🇸 Interactive Exploration

💻 Libraries

📊 Objects in Orbit Through Time

🇺🇸 **Objects by State**

📊 Most Populated Orbits

🇺🇸 'Mass' Column Study

📄 ☒ Research into 'Mass' Outliers

📄 ☒ Research into 'Mass' Values Depending on 'OpOrbits'

📄 ☒ Exploration of 'Mass' in Deeper Detail

📄 Machine Learning - 'Mass' Regression

💻 Libraries

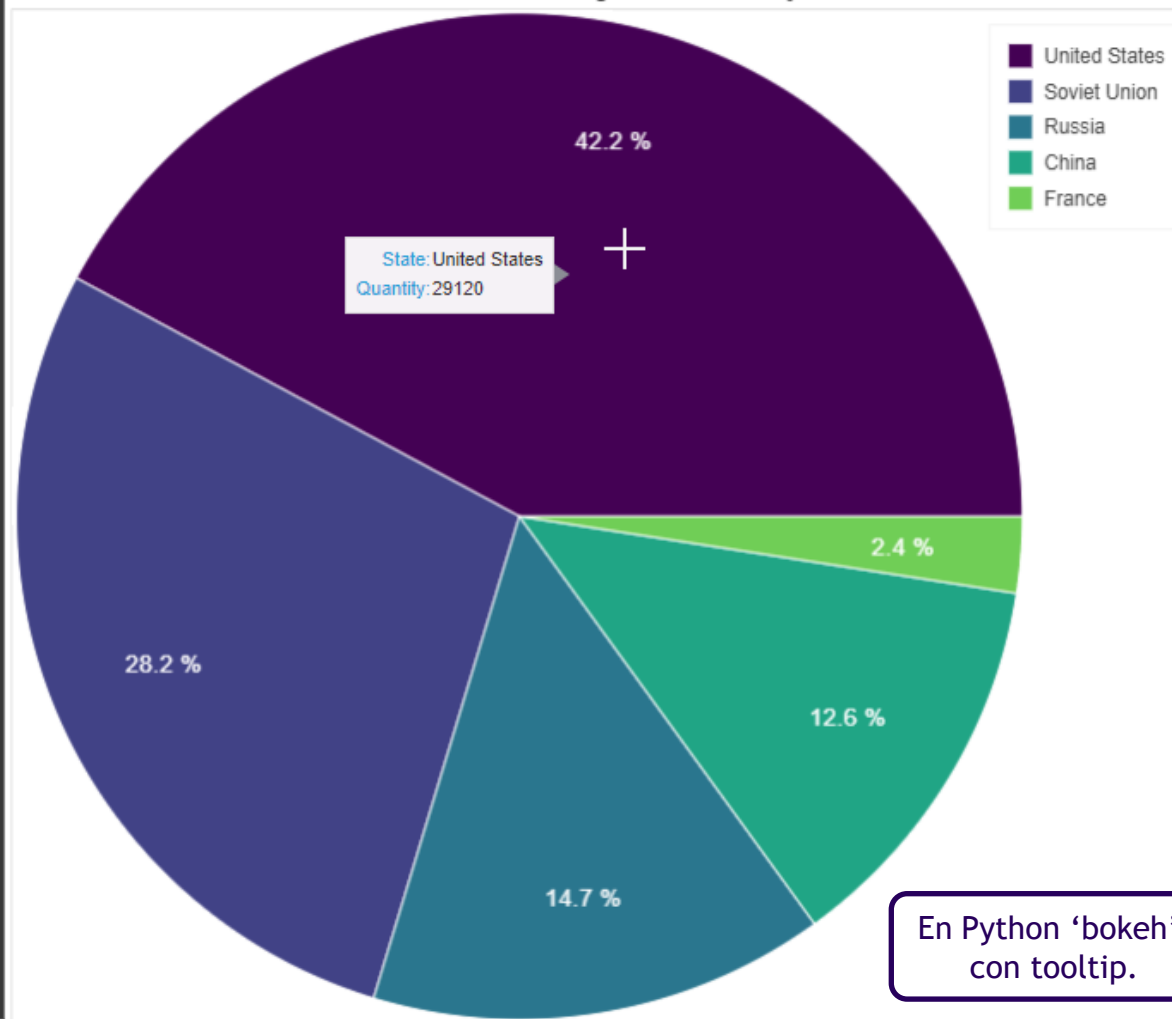
+ Code + Text

Connect ▾

🔗 Gemini



Five Countries That Originated the Most Objects



En Python 'bokeh',
con tooltip.



Objects in Orbit Through Time

Objects by State

Most Populated Orbits

'Mass' Column Study

☒ Research into 'Mass' Outliers

☒ Research into 'Mass' Values Depending on 'OpOrbits'

☒ Exploration of 'Mass' in Deeper Detail

Machine Learning - 'Mass' Regression

Libraries

k Nearest Neighbors Regression

kNN Optimization

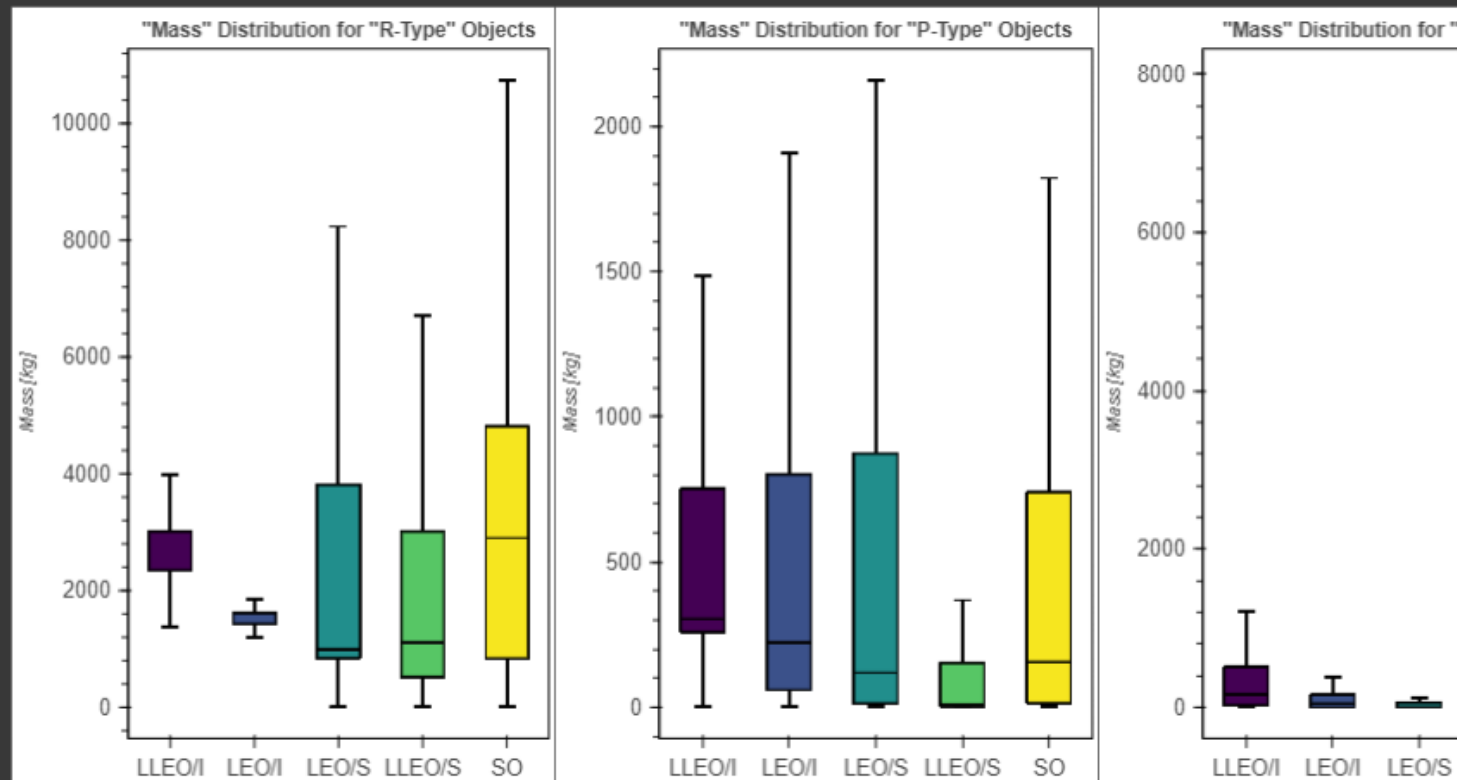
Random Forest Regression

RDF Optimization

Gradient Boosting Regression

```
# Faceted plots
ncols = df_mass_orb_type['Type_First_Letter'].nunique()
grid = gridplot(plots, ncols = ncols)
# Visualization
show(grid)
```

Gráfico facetado en Python
'bokeh', sin atípicos.





📁 k Nearest Neighbors
Regression

🔦 kNN Optimization

📁 Random Forest Regression

🔦 RDF Optimization

📁 Gradient Boosting
Regression

🔦 GBM Optimization

📁 Support Vector Machine
Regression

🔦 SVM Optimization

📁 Extreme Gradient Boosting
Regression

🔦 XGB Optimization

🔥 Model Selection

💰 Monetization

💡 Insights / Conclusions

🔮 Future Directions

+ Section

```
[ ] # Printing the DataFrame
ml_results_df_ordered = ml_results_df.sort_values(by = ['Final Adjusted R2'], ascending = False)
print(
    ml_results_df_ordered[
        ['Model', 'Final Adjusted R2', 'Final RMSE', 'Final RAE', 'Best R2 Score', 'Best Parameters']
    ].map(lambda x: list(x.values()) if isinstance(x, dict) else x)
)
```

	Model	Final Adjusted R2	Final RMSE	Final RAE	Best R2 Score	Best Parameters
1	RDF Reg	0.982490	906.689630	0.051243	0.965414	[25, sqrt, 1, 2, 380]
2	GBM Reg	0.981033	943.649359	0.051275	0.968199	[25, sqrt, 2, 5, 160]
3	GBM Reg	0.977067	1041.489312	0.051538	0.981108	[None, sqrt, 1, 10, 380]
5	GBM Reg	0.977067	1041.489633	0.051538	0.981108	[None, sqrt, 1, 10, 280]
4	GBM Reg	0.977067	1041.493103	0.051539	0.981108	[None, sqrt, 1, 10, 220]
11	XGB Reg	0.973169	1126.521971	0.054162	0.978444	[0.8, 0.1, None, 360, 1.0]
10	XGB Reg	0.973075	1128.500991	0.054885	0.978451	[0.8, 0.1, None, 320, 1.0]
9	XGB Reg	0.973049	1129.028778	0.055241	0.978447	[0.8, 0.1, None, 300, 1.0]
0	kNN Reg	0.933780	1763.226547	0.179632	0.932780	[euclidean, 5, distance]
6	SVM Reg	0.703453	3745.144158	0.260710	0.893006	[10, linear]
8	SVM Reg	0.647048	4085.823710	0.356155	0.872526	[50, 3, auto, poly]
7	SVM Reg	0.602688	4334.980719	0.464197	0.839854	[10, 3, auto, poly]

```
[ ] # Parameter names ordered as in the last printout
for index, row in ml_results_df_ordered.iterrows():
    model = row['Model']
    best_parameters = row['Best Parameters']
    print(model, list(best_parameters.keys()))
```

```
RDF Reg ['max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', 'n_estimators']
GBM Reg ['max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', 'n_estimators']
GBM Reg ['max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', 'n_estimators']
GBM Reg ['max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', 'n_estimators']
GBM Reg ['max_depth', 'max_features', 'min_samples_leaf', 'min_samples_split', 'n_estimators']
XGB Reg ['colsample_bytree', 'learning_rate', 'max_depth', 'n_estimators', 'subsample']
XGB Reg ['colsample_bytree', 'learning_rate', 'max_depth', 'n_estimators', 'subsample']
XGB Reg ['colsample_bytree', 'learning_rate', 'max_depth', 'n_estimators', 'subsample']
kNN Reg ['metric', 'n_neighbors', 'weights']
```

Métricas desde Python
'scikit-learn' tras
búsquedas en cuadrícula.

Data Science III: NLP & Deep Learning aplicado a Ciencia de Datos

Juan Morales Volosín

Ha realizado y completado con éxito su curso en Coderhouse.
La duración fue de 16 horas dictadas a lo largo de 4 semanas,
cumpliendo todos los requisitos académicos exigidos.

24 de octubre de 2024



Tamara Drajner

Coderhouse
Head de operaciones académicas



Christian Patiño

CEO y Co-founder en Coderhouse



Reconocimiento de Estilo de Escritura para SpaceNews.com

Instrucciones

Objetivo

Contexto de Negocio

Contexto Analítico

Recursos

Fuente

Bibliotecas

Configuraciones

Funciones Personalizadas

Descargas

Carga de Datos y Exploración

Punto de Control

Exploración Visual

Preprocesamiento para NLP

Estandarización



Reconocimiento de Estilo de Escritura para SpaceNews.com

Curso:

Data Science III: NLP & Deep Learning Aplicado a Ciencia de Datos

Comisión #60960

Alumno:

Morales Volosín, Juan Ignacio

Profesor:

Russo Locati, Ignacio

Tutor:

Alric, Juan Cruz

Instrucciones

Tareas realizadas principalmente con Python 'NLTK' y 'PyTorch'.

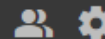
Table of contents



+ Code + Text

Connect

+ Gemini



Stopwords

Lematización

Stemming y Punto de
Control

Comparación
Antes/Después de
Preprocesamiento

Tareas de NLP

Bag of Words

BoW desde Lemas

BoW desde Raíces

Vectorización TF-IDF

TF-IDF desde Lemas

TF-IDF desde Raíces

Punto de Control

Aprendizaje Automático

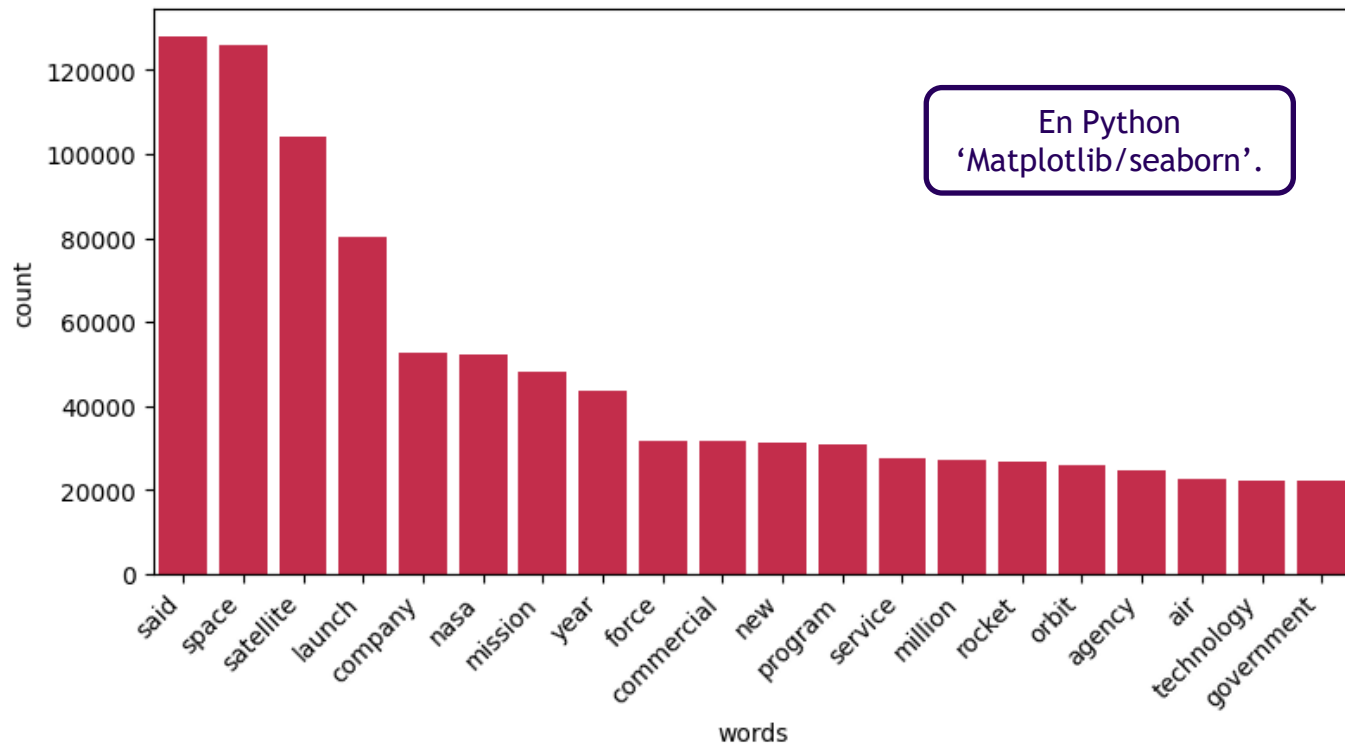
Preparación

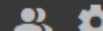
Regresión Logística

```
gr_wcount.fig.suptitle('Los 20 Lemas Más Frecuentes de los Todos los Artículos', y = 1.05)  
gr_wcount.set_xticklabels(rotation = 45, ha = 'right')  
gr_wcount.axes[0, 0].spines['top'].set_visible(True)  
gr_wcount.axes[0, 0].spines['right'].set_visible(True)  
plt.show()
```



Los 20 Lemas Más Frecuentes de los Todos los Artículos





XGBoost

Desde Lemas

Desde Raíces

Punto de Control

Aprendizaje Profundo

MLP desde Lemas

MLP desde Raíces

Punto de Control

Selección del Modelo

Decisión

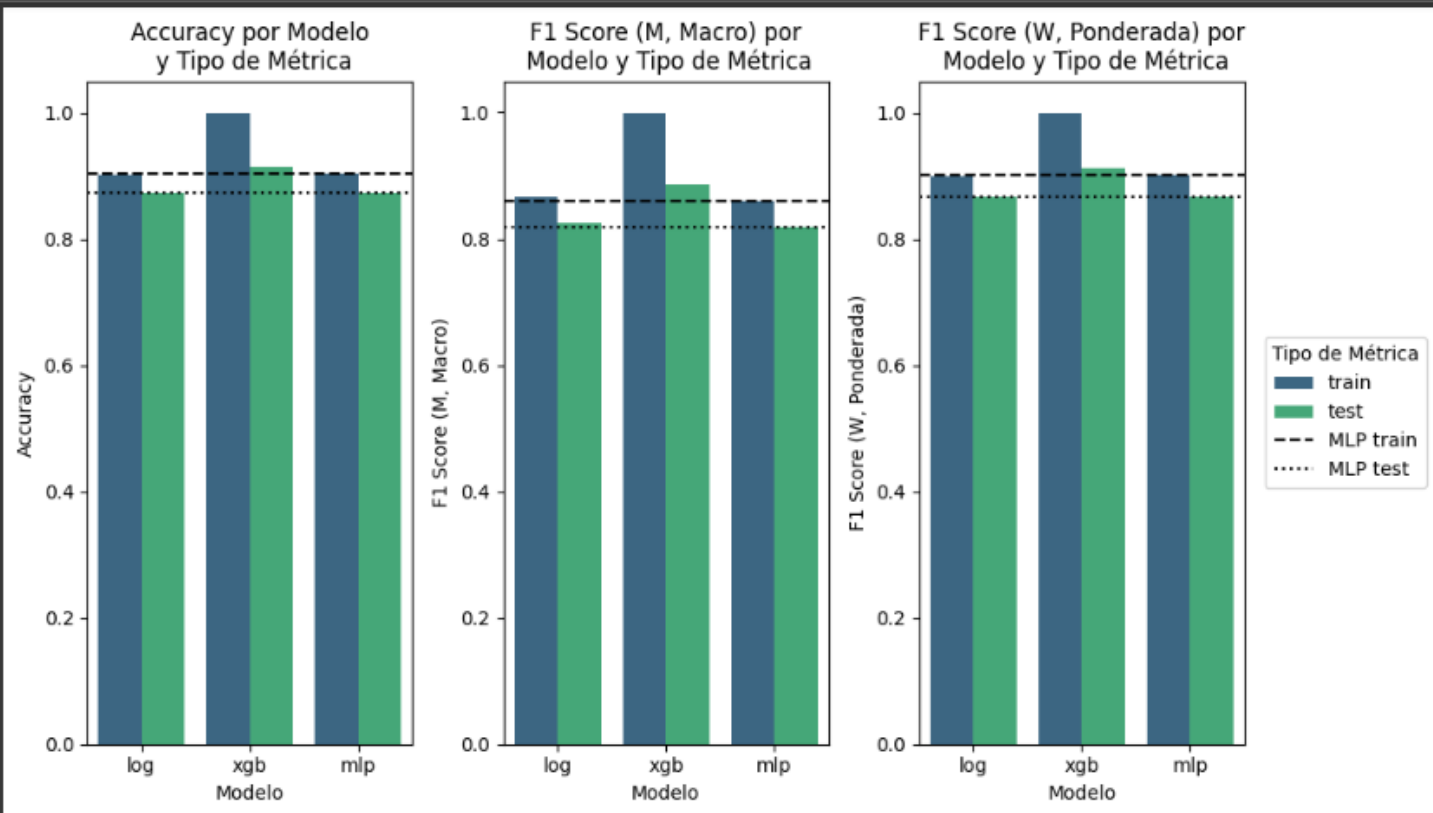
Conclusiones

NLP

Redes Neuronales

Líneas Futuras

+ Section



Mediante el gráfico, se refuerzan las ideas:

- XGBoost presenta mayores valores siempre, pero más tendencia al sobreajuste.
- Regresión logística y MLP (la red neuronal) presentan valores muy parejos entre sí, y mucho menor tendencia al sobreajuste.

Gráfico facetado en Python
'matplotlib/seaborn'.